

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平8-212083

(43)公開日 平成8年(1996)8月20日

(51)Int.Cl.<sup>6</sup>

G 0 6 F 9/46

識別記号

3 1 3 C

3 1 0 L

庁内整理番号

F 1

技術表示箇所

審査請求 未請求 請求項の数4 O L (全 15 頁)

(21)出願番号 特願平7-19383

(22)出願日 平成7年(1995)2月7日

(71)出願人 000000295

沖電気工業株式会社

東京都港区虎ノ門1丁目7番12号

(72)発明者 河井 淳

東京都港区虎ノ門1丁目7番12号 沖電気  
工業株式会社内

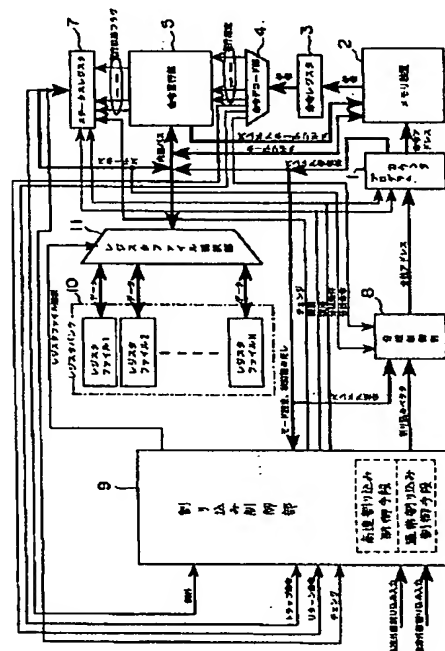
(74)代理人 弁理士 佐々木 宗治 (外3名)

(54)【発明の名称】 割り込み処理装置

(57)【要約】

【目的】 割り込み処理を行う際のレジスタ内容の退避及び復帰処理に要する時間が短縮され、割り込み処理性能が向上した割り込み処理装置。

【構成】 複数のレジスタファイル1～Nから構成されるレジスタバンク10と、プロセッサに係る状態情報を保持するステータスレジスタ7と、複数の通常割り込み信号及び複数の高速割り込み信号を入力し、それぞれの割り込み処理プログラムへの分岐及び復帰を制御する割り込み制御部9とを備えた割り込み処理装置であって、前記割り込み制御部9は、割り込み信号の入力時における状態レジスタの保持情報及びプログラム戻りアドレスの退避、割り込みベクタの生成による分岐、及び割り込み終了後の退避情報の復帰を行うと共に、高速割り込み処理が可能な場合には、新規レジスタファイルの割り付け後に生成する割り込みベクタにより分岐し、割り込み終了後は分岐前に使用していたレジスタファイルに戻した後に退避情報の復帰を行う通常及び高速割り込み制御手段を有するもの。



本発明の実施例による割り込み処理装置の構成を示す図

【特許請求の範囲】

【請求項1】 複数のレジスタファイルから構成されるレジスタバンクと、  
プロセッサに係る状態情報を保持する状態レジスタと、  
複数の通常割り込み信号及び複数の高速割り込み信号を入力し、それぞれの割り込み処理プログラムへの分岐及び復帰を制御する割り込み制御手段とを備えた割り込み処理装置であって、  
前記割り込み制御手段は、  
前記通常割り込み信号が入力された場合に、その時点における前記状態レジスタの保持情報及びプログラム戻りアドレスを退避保存し、該当通常割り込み信号により一義的に与えられる割り込みベクタから所定の割り込み処理プログラムに分岐させ、この分岐した割り込み処理プログラムの実行終了後に、前記退避保存した状態レジスタの保持情報を状態レジスタに復帰させると共に退避保存したプログラム戻りアドレスから分岐前に実行中のプログラムに復帰させる通常割り込み制御手段と、  
前記高速割り込み信号が入力された場合に、その時点における前記状態レジスタの保持情報及びプログラム戻りアドレスを退避保存し、次に前記レジスタバンク内に使用中でないレジスタファイルの有無を判別して有る場合には、その時点で使用中のレジスタファイルの代りにレジスタバンク内の使用中でないレジスタファイルを割り付け、該当高速割り込み信号により一義的に与えられる割り込みベクタから所定の割り込み処理プログラムに分岐させ、この分岐した割り込み処理プログラムの実行終了後に、分岐前に使用していたレジスタファイルに戻し、前記退避保存した状態レジスタの保持情報を状態レジスタに復帰させると共に退避保存したプログラム戻りアドレスから分岐前に実行中のプログラムに復帰させ、また前記レジスタバンク内に使用中でないレジスタファイルの有無を判別して無い場合には、レジスタファイルの新規割り付けは行わずに、前記通常割り込み制御手段による制御を行わせるための別の割り込みベクタで指定される割り込み処理プログラムに分岐させる高速割り込み制御手段とを有することを特徴とする割り込み処理装置。

【請求項2】 前記割り込み制御手段は、  
複数の各高速割り込みイネーブルフラグを含む、各種割り込みイネーブルフラグにより構成されるモードレジスタと、  
前記複数の高速割り込み信号及び通常割り込み信号を含む複数の割り込み要因信号、並びに前記モードレジスタの保持情報を入力し、複数の割り込み要因信号のうちの最高優先度の信号を選択出力する優先度判定手段と、  
前記レジスタバンク内における使用中のレジスタファイルの数を計数し、レジスタファイル選択信号を出力する加減算カウンタと、  
前記優先度判定手段の出力、モードレジスタの保持情報

及び加減算カウンタの出力に基づき、分岐処理用の割り込みベクタを生成するベクタ生成手段と、  
前記状態レジスタの保持情報、優先度判定手段の出力及び復帰命令に基づき、前記退避保存、復帰及び状態レジスタの保持情報の変更の各制御信号を出力するシーケンス制御手段とを備えたことを特徴とする請求項1記載の割り込み処理装置。

【請求項3】 前記割り込み制御手段に含まれる加減算カウンタが出力するレジスタファイル選択信号によって、前記レジスタバンク内の複数のレジスタファイルのうちの1つを選択して内部バスに接続させるレジスタファイル選択手段を備えた請求項1又は請求項2記載の割り込み処理装置。

【請求項4】 前記状態レジスタは、プロセッサに係る状態情報の一部として、前記レジスタバンク内の複数の各レジスタファイルが使用中であるか否かを示すフラグ情報を含むことを特徴とする請求項1ないし請求項3のいずれかに記載の割り込み処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は高速割り込み処理手段を備えた割り込み処理装置に関するものである。

【0002】

【従来の技術】図8は従来の計算機の割り込み処理装置の構成を示す図である。図8において、1はプログラムカウンタ、2はメモリ装置、3は命令レジスタ、4は命令デコード部、5は命令実行部、6はレジスタファイル、7Aは従来のステータスレジスタ、8は分岐制御部、9Aは従来の割り込み制御部である。図9は図8の装置の割り込み処理の開始、実行及び復帰順序を示す流れ図である。

【0003】図9を参照し、図8の動作を説明する。図8において、プログラムカウンタ1は次に実行すべき命令アドレスを生成する部分で、ここから命令アドレスがメモリ装置2に与えられる。メモリ装置2は与えられた命令アドレスから実行すべき命令コードを読みだし、命令レジスタ3に格納する。命令レジスタ3に格納された命令コードは命令デコード部4にてデコード（解読）され命令実行部5に与えられる制御信号に変換される。命令実行部5は与えられた命令実行指定のための制御信号に従い処理を行なう。このとき、処理される対象となるオペランドデータ（演算の対象となるデータ）はレジスタファイル6から内部バスを介して読みだす。また、処理結果データは同様に内部バスを介して命令実行部5からレジスタファイル6内の指定レジスタに格納される。レジスタファイル6内のレジスタ指定は命令コードの中に表わされる。これと同時に、命令実行結果を反映する実行結果フラグはステータスレジスタ7Aに格納される。通常の命令実行はこのようにして行なわれる。

【0004】実行すべき命令がメモリアクセス命令の場合

合には、命令実行部5ではメモリアドレス計算が行なわれる。メモリアクセス命令がメモリ読みだしの場合には、計算されたメモリアドレスをメモリ装置2に与えることでメモリアクセスが行なわれる。メモリから読みだされたデータは、内部バスを介してレジスタファイル6内の命令コードで指定されるレジスタに格納される。また、メモリアクセス命令がメモリ書き込みの場合には、命令実行部5にて計算されたメモリアドレス、および、レジスタファイル6内の命令コードで指定されるレジスタから読みだしたメモリ書き込みデータを、内部バスを介してメモリ装置2に与えることでメモリアクセスが行なわれ、所定のメモリアドレスにデータが書き込まれる。これと同時に命令実行結果を反映する実行結果フラグは、ステータスレジスタ7に格納される。通常の命令実行はこのようにして行なわれる。

【0005】実行すべき命令が分岐命令の場合には、命令実行部5では分岐先命令アドレスが計算される。計算された分岐アドレスは分岐制御部8に与えられる。同時に命令デコード部4からは分岐制御のための分岐命令信号が、ステータスレジスタ7Aからは分岐条件が、それぞれ分岐制御部8に与えられる。分岐制御部8ではこれらの入力信号から分岐条件が成立するか否かの判定を行ない、分岐条件が成立する場合には次に実行すべき分岐アドレスをプログラムカウンタ1に与える。

【0006】実行すべき命令がトラップ命令（指定されたアドレスへの、予定されていない条件付き飛越してあって、ハードウェアにより自動的に起動され、飛越してきた元の場所が記録される命令）であった場合、直前に実行した命令がオーバフロー、アンダフロー、ゼロ除算などの例外条件を発生した場合、または外部割り込み入力を与えられた場合のいずれかの場合には割り込み要求が発生したことになる。この場合にはその時点で実行中のプログラム実行は一時中断され、各割り込み要因に対応する割り込み処理を行なう。

【0007】この割り込み処理を実現するには、図9の(a)のステップ（以下単にSと記す）51において、まず前記例外条件、トラップ命令、および外部割り込み入力割り込み制御部9Aに与えられ受け付けられる。割り込み制御部9AはS52で、これらの割り込み要因のうち、まず割り込みを禁止されている要因を除く。そして残った要因の中から最も優先度の高いものを選択する。前記の選択の結果として割り込みが成立した場合には、割り込み制御部9Aは、まずS53で、プログラムカウンタ1に退避信号を与え、割り込まれた命令アドレスを内部バスを介してレジスタファイル6の特定レジスタに退避する。

【0008】次にS54で、退避信号をステータスレジスタ7Aに与え、割り込まれた時点のステータスレジスタ7Aの内容を、同様に内部バスを介してレジスタファイル6の特定レジスタに退避する。その後、割り込み制

御部9AはS55で、選択された割り込み要因に対応する割り込みベクタ（割り込み信号を、その値によってそれぞれ対応するプログラムに振り分ける方式をベクタ割り込みといい、この場合の割り込みベクタとは、対応するプログラムを選択するデータの意である。）を分岐制御部8に出力する。分岐制御部8では与えられた割り込みベクタから実際の割り込み処理ルーチンの先頭アドレスを計算しプログラムカウンタ1に出力する。これ以降は通常の命令実行と同様にプログラムカウンタ1から与えられる命令アドレスでメモリ装置2から実行すべき命令を読みだし実行を行なう。

【0009】上記のシーケンスを経て割り込み処理ルーチンに命令実行が移行する。割り込み処理ルーチンでは、通常、割り込み制御部9Aは、図9の(b)のS61において、まず最初に割り込み時点のレジスタファイル6の内容をメモリ装置2に退避する。これは割り込み処理プログラム中の命令として実行される。これにより割り込み処理でレジスタファイル6を使用しても、割り込み以前のレジスタファイル6の内容は保存されることになる。それからS62において、本来の割り込み処理を実行する。割り込み処理が終了した場合には、割り込み制御部9AはS63において、先程のS61とは逆にメモリ装置2に退避してある割り込まれた時点でのレジスタファイル6の内容をレジスタファイル6に復帰させる。この処理も割り込み処理プログラム中の命令として実行される。そして最後に割り込み処理から復帰するために割り込みリターン命令を実行する。

【0010】割り込みリターン命令は、命令デコード部4からリターン命令制御信号が割り込み制御部9Aに与えられ処理がはじまる。割り込み処理制御部9Aは図9の(c)のS71において、まずステータスレジスタ7Aに復帰信号を与え、レジスタファイル6の特定レジスタに退避されている割り込み時点のステータスレジスタ7Aの内容を、内部バスを介して読みだしステータスレジスタ7Aに復帰する。次にS72において、復帰信号をプログラムカウンタ1に与え、レジスタファイル6の特定レジスタに退避されている割り込まれた命令アドレスを読みだし、プログラムカウンタ1に復帰する。プログラムカウンタ1からは復帰後の最初の命令アドレスがメモリ装置2に与えられ、割り込みにより中断されたプログラムの実行が再開される。

【0011】

【発明が解決しようとする課題】しかしながら上記のような計算機の割り込み処理装置では、レジスタファイルが一组しかないため、割り込み処理の開始時にレジスタファイルの内容をメモリ装置に退避し、また、割り込み処理の終了時に退避したレジスタファイルの内容をメモリ装置から復帰する必要があった。このレジスタファイルの内容の退避、および復帰処理は命令実行により行なわれる場合と、ハードウェアにより行なわれる場合の2

通りの場合がある。いずれの場合にも、この処理のために数十命令の実行時間に相当する時間が退避、復帰、それぞれに費やされるため、割り込み応答時間、および割り込み処理時間が長くなり性能低下をもたらしているという問題点があった。特に高速の割り込み応答処理が要求される場合には、上記のレジスタファイルの内容の一部のみを退避、および復帰し、限られた数のレジスタを使用して割り込み処理を行なうこともある。しかし、この場合にもレジスタの退避、および復帰の時間は必要である。また、削減された数のレジスタを使用しているプログラミングでは、割り込み処理プログラム記述に大きな制限を与えることになり、高級言語での割り込み処理プログラミングや、移植性の高い割り込み処理プログラムとすることが著しく困難になる。少数のレジスタのみを使用したプログラムのためかえって性能低下をもたらす場合もある。

【0012】また、上記の問題を回避するために、割り込み処理などのために特別なレジスタセットを用意している計算機もある。これによりレジスタファイルの退避、および、復帰を行なうことによる問題を軽減することが出来る。しかしながら、用意される特別なレジスタセットの組数は有限であり、割り込みが多重に発生するような全ての状況においてレジスタファイルの退避、および、復帰を省くことは保証されない。従って、このような計算機においては、毎割り込み受け付け時にソフトウェアにより特別なレジスタセットを使用可能か否かの判定を行ない、可能な場合にのみレジスタファイルの退避、および、復帰を省いている。仮にほとんど割り込み発生時に特別なレジスタセットを使用して割り込み処理を実行することが出来たとしても、それを使用可能か否かの判定はソフトウェアにより必ず実行されなければならない、そのための所要時間が性能低下をもたらしているという問題点もあった。

#### 【0013】

【課題を解決するための手段】本発明に係る割り込み処理装置は、複数のレジスタファイルから構成されるレジスタバンクと、プロセッサに係る状態情報を保持する状態レジスタと、複数の通常割り込み信号及び複数の高速割り込み信号を入力し、それぞれの割り込み処理プログラムへの分岐及び復帰を制御する割り込み制御手段とを備えた割り込み処理装置であって、前記割り込み制御手段は、前記通常割り込み信号が入力された場合に、その時点における前記状態レジスタの保持情報及びプログラム戻りアドレスを退避保存し、該当通常割り込み信号により一義的に与えられる割り込みベクタから所定の割り込み処理プログラムに分岐させ、この分岐した割り込み処理プログラムの実行終了後に、前記退避保存した状態レジスタの保持情報を状態レジスタに復帰させると共に退避保存したプログラム戻りアドレスから分岐前に実行中のプログラムに復帰させる通常割り込み制御手段と、

前記高速割り込み信号が入力された場合に、その時点における前記状態レジスタの保持情報及びプログラム戻りアドレスを退避保存し、次に前記レジスタバンク内に使用中でないレジスタファイルの有無を判別して有る場合には、その時点で使用中のレジスタファイルの代りにレジスタバンク内の使用中でないレジスタファイルを割り付け、該当高速割り込み信号により一義的に与えられる割り込みベクタから所定の割り込み処理プログラムに分岐させ、この分岐した割り込み処理プログラムの実行終了後に、分岐前に使用していたレジスタファイルに戻し、前記退避保存した状態レジスタの保持情報を状態レジスタに復帰させると共に退避保存したプログラム戻りアドレスから分岐前に実行中のプログラムに復帰させ、また前記レジスタバンク内に使用中でないレジスタファイルの有無を判別して無い場合には、レジスタファイルの新規割り付けは行わずに、前記通常割り込み制御手段による制御を行わせるための別の割り込みベクタで指定される割り込み処理プログラムに分岐させる高速割り込み制御手段とを有するものである。

#### 【0014】

【作用】本発明に係る割り込み処理装置は、複数のレジスタファイルから構成されるレジスタバンクと、プロセッサに係る状態情報を保持する状態レジスタと、複数の通常割り込み信号及び複数の高速割り込み信号を入力し、それぞれの割り込み処理プログラムへの分岐及び復帰を制御する割り込み制御手段とを備えており、前記割り込み制御手段は通常割り込み制御手段と高速割り込み制御手段とを有する。通常割り込み制御手段は、前記通常割り込み信号が入力された場合に、その時点における前記状態レジスタの保持情報及びプログラム戻りアドレスを退避保存し、該当通常割り込み信号により一義的に与えられる割り込みベクタから所定の割り込み処理プログラムに分岐させ、この分岐した割り込み処理プログラムの実行終了後に、前記退避保存した状態レジスタの保持情報を状態レジスタに復帰させると共に退避保存したプログラム戻りアドレスから分岐前に実行中のプログラムに復帰させる。高速割り込み制御手段は、前記高速割り込み信号が入力された場合に、その時点における前記状態レジスタの保持情報及びプログラム戻りアドレスを退避保存し、次に前記レジスタバンク内に使用中でないレジスタファイルの有無を判別して有る場合には、その時点で使用中のレジスタファイルの代りにレジスタバンク内の使用中でないレジスタファイルを割り付け、該当高速割り込み信号により一義的に与えられる割り込みベクタから所定の割り込み処理プログラムに分岐させ、この分岐した割り込み処理プログラムの実行終了後に、分岐前に使用していたレジスタファイルに戻し、前記退避保存した状態レジスタの保持情報を状態レジスタに復帰させると共に退避保存したプログラム戻りアドレスから分岐前に実行中のプログラムに復帰させ、また前記レジ

スタバンク内に使用中でないレジスタファイルの有無を判別して無い場合には、レジスタファイルの新規割り付けは行わずに、前記通常割り込み制御手段による制御を行わせるための別の割り込みベクタで指定される割り込み処理プログラムに分岐させる。従って本発明においては、高速割り込み処理が可能な場合には、その時点で使用中のレジスタファイルをハードウェアにより高速で切り替え高速割り込み処理プログラムに分岐し、この分岐したプログラムの実行終了後に、再び分岐前に使用していたレジスタファイルに高速で戻して分岐前のプログラムに復帰させるので、従来方式のようにレジスタファイルの内容のメモリ装置への退避処理及び復帰処理が不要となり、従来よりも割り込み処理の事前処理及び事後処理時間が大幅に低減され、割り込み処理性能が向上する。また本発明では、高速割り込み処理時にレジスタファイルそのものを切り替えるので、レジスタファイル内に含まれるレジスタの数は非割り込み処理である通常処理と同数であり、その結果、従来方式におけるレジスタファイルの一部のみを退避及び復帰させ、限られた数のレジスタを使用して割り込み処理を行う場合に生じる割り込み処理プログラム記述上の制限が全くない。

#### 【0015】

【実施例】図1は本発明の実施例による割り込み処理装置の構成を示す図である。図1において、1～5及び8は図8と同一のものであり、7は本発明によるステータスレジスタである。一般にステータスレジスタはプロセッサに係る状態情報（例えば命令実行結果フラグ等）を保持するものであるが、本発明のステータスレジスタ7は、この一般のプロセッサに係る状態情報のほかに、レジスタバンク内の複数の各レジスタファイルが使用中であるか否かを示すフラグ情報（この例ではチェンジフラグと呼び詳細は図4で説明する）も含むものである。9は本発明による割り込み制御部であり、複数の高速割り込み信号と複数の通常割り込み信号とを入力し、それぞれの割り込み処理への分岐及び復帰を制御する割り込み制御手段である。従って割り込み制御部9は、機能的に、高速割り込み制御手段と通常割り込み制御手段を含むものである。10は複数N個のレジスタファイル1～Nを含むレジスタバンクであり、各レジスタファイルが使用中であるか否かを示すフラグ情報は前記ステータスレジスタ7から読み取り可能になっている。11はレジスタファイル選択部であり、割り込み制御部9から供給されるレジスタファイル選択信号によって指定される1つのレジスタファイルを選択して内部バスとの接続を行う。

【0016】図1の動作を説明する。図1において、プログラムカウンタ1は次に実行すべき命令アドレスを生成する部分で、ここから命令アドレスがメモリ装置2に与えられる。メモリ装置2は与えられた命令アドレスから実行すべき命令コードを読みだし、命令レジスタ3に

格納する。命令レジスタ3に格納された命令コードは、命令コード部にてデコードされ命令実行部5に与えられる制御信号に変換される。命令実行部5は与えられた命令実行指定のための制御信号に従い処理を行なう。このとき、処理される対象となるオペランドデータは、レジスタバンク10内にあるレジスタファイル1～Nの中の特定のレジスタファイルの指定されるレジスタから内部バスを介して読みだす。この特定のレジスタファイルは、割り込み制御部9から与えられるレジスタファイル選択信号により選択され、内部バスに接続される。この特定のレジスタファイル内のレジスタ指定は命令コードの中に表わされる。また、処理結果データは、同様に内部バスを介して命令実行部5から上記の特定のレジスタファイル内の指定されるレジスタに格納される。これと同時に、命令実行結果を反映する実行結果フラグはステータスレジスタ7に格納される。通常の命令実行はこのようにして行なわれる。

【0017】実行すべき命令がメモリアクセス命令の場合には、命令実行部5ではメモリアドレス計算が行なわれる。メモリアクセス命令がメモリ読みだしの場合には、計算されたメモリアドレスをメモリ装置2に与えることでメモリアクセスが行なわれる。メモリから読みだされたデータは、内部バスを介してレジスタファイル内の命令コードで指定されるレジスタに格納される。また、メモリアクセス命令がメモリ書き込みの場合には、命令実行部5にて計算されたメモリアドレス、および、レジスタファイル内の命令コードで指定されるレジスタから読みだしたメモリ書き込みデータを、内部バスを介してメモリ装置2に与えることでメモリアクセスが行なわれ、所定のメモリアドレスにデータが書き込まれる。

【0018】実行すべき命令が分岐命令の場合には、命令実行部5では分岐先命令アドレスが計算される。計算された分岐アドレスは分岐制御部8に与えられる。同時に、命令デコード部4からは分岐制御のための分岐命令信号が、ステータスレジスタ7からは分岐条件が、それぞれ分岐制御部8に与えられる。分岐制御部8ではこれらの入力信号から分岐条件が成立するか否かの判定を行ない、分岐条件が成立する場合には次に実行すべき分岐アドレスをプログラムカウンタ1に与える。実行すべき命令がトラップ命令であった場合、直前に実行した命令がオーバフロー、アンダフロー、ゼロ除算などの例外条件を発生した場合、または高速外部割り込み入力あるいは通常外部割り込み入力が与えられた場合には、いずれの場合にも割り込み要求が発生したと認識される。これらの場合にはその時点で実行中のプログラム実行は一時中断され、各割り込み要因に対応する割り込み処理を行なう。

【0019】図2は図1の割り込み制御部9の具体的な構成例を示す図であり、図3は図2のモードレジスタ22の構成を示す図であり、図4は図1のレジスタファイ

ル1～Nの使用法を説明する図であり、図5は本発明の実施例による割り込み開始処理の流れ図であり、図6は本発明の実施例による高速及び通常割り込み処理の流れ図であり、図7は本発明の実施例により割り込み復帰処理の流れ図である。図2において、21は優先度判定ブロック、22はモードレジスタ、23はN進アップダウンカウンタ、24はシーケンサ、25はベクタ生成部、26、27はアンプ、28は丸印の付加された入力端の信号の反転信号と他方の入力端の信号との論理積を出力するANDゲートである。

【0020】図1～図4を参照し、図5～図7の流れ図に従い、本発明の実施例による割り込み処理について説明する。本発明の実施例による割り込み処理は、まず図5のS1において、例外条件、トラップ命令、高速外部割り込み入力、および通常外部割り込み入力が入力された図1の割り込み制御部9に受け付けられる。割り込み制御部9はS2で、これらの割り込み要因入力に対して割り込み受け付け可否の判定と、複数の受け付け可能な割り込み入力に対しては優先度の判定を行う。そして上記の判定は図2の優先度判定ブロック21によって行なわれる。

【0021】優先度判定ブロック21では、入力された各割り込み要因に対して、それぞれ割り込み受け付け可能か否かのチェックが行なわれる。これは優先度判定ブロック21内にある割り込みマスクレジスタ、および、割り込み可能レベルレジスタの内容に照らし合わせて判定される。すなわち、予めプログラムにより設定された個々の割り込み要因毎の割り込みマスクフラグが“0”であり、かつ、割り込み要求が与えられた時点での割り込み可能レベルよりも高い場合にのみ、当該割り込み要因は受け付けられる。割り込み可能レベルは、計算機の処理のレベルを逐次反映するもので、ある時点に計算機が既にあるレベルの割り込み処理中である場合には、割り込み可能レベルはその時点に処理している割り込み処理にたいする要因よりも高い場合にのみ受け付け可能となる。さらに、同時に複数の割り込み受け付け可能な割り込み要求が与えられた場合には、それらのうち一番優先度の高いものが選択される。割り込み優先度は、一般的にハードウェアにより予め定められていて優先度判定ブロック21内に結線論理（固定優先論理）として組み込まれている。

【0022】前記の優先度判定による選択の結果として割り込みが成立した場合には、割り込み制御部9は、まずS3で、プログラムカウンタ1に退避信号を与え、割り込まれた命令アドレスを内部バスを介してその時点で使用中のレジスタファイルの特定レジスタに退避する。次にS4で、退避信号をステータスレジスタ7に与え、割り込まれた時点のステータスレジスタ7の内容を同様に内部バスを介してその時点で使用中のレジスタファイルの特定レジスタに退避する。この後本発明の実施例で

は、上記の動作により選択された割り込み要因にたいして、高速割り込み処理を行なうものと通常割り込み処理を行なうものとを判定し、それぞれ別の動作を行なう。優先度判定ブロック21にて選択された割り込み要因が、トラップ命令、例外条件、あるいは高速割り込み入力のいずれかの場合は、図2のモードレジスタ22の設定状態に応じて高速割り込み処理を行う。

【0023】図3には、図2のモードレジスタ22の構成が示されており、モードレジスタ22は、図3に示す高速割り込みイネーブル1～N、高速例外処理イネーブル、高速トラップ処理イネーブル、および高速割り込みイネーブルの各フラグで構成される。上記4種類のイネーブルフラグのうち、高速割り込みイネーブルは、マスターイネーブルで、このフラグが“1”のときのみ他のフラグが有効となる。したがって、高速割り込みイネーブルが“0”のときは、どんな割り込み要求条件に対しても高速割り込み処理は行なわれない。高速割り込みイネーブル1～Nは高速外部割り込み入力1～Nにそれぞれ対応していて、高速外部割り込み入力が与えられたとき、これに対応する高速割り込みイネーブルフラグが“1”である場合に、該当高速割り込み処理が行なわれる。以下、高速例外処理イネーブルは例外信号が与えられた場合、また、高速トラップ処理イネーブルはトラップ命令が与えられた場合、高速割り込みイネーブル1～Nと同様のはたらきをする。

【0024】優先度判定ブロック21は、図5のS5及びS6において、入力された割り込み要因が高速割り込み処理を要求する要因であるか否か及びモードレジスタ22の設定値が与える条件を満足するか否かを判別する。そしてS5及びS6の判別結果が共にYESの場合に、割り込み制御部9はS7において、レジスタバンク10内に使用中でないレジスタファイルが有るか否か、即ちレジスタファイルがアベイラブルか否かを判別し、この判別結果がYESの場合には高速割り込み処理を行なう。これは本発明による高速割り込み処理においては、割り込まれた時点に使用していたレジスタファイルから新たな未使用のレジスタファイルに切り替えて割り込み処理を行なうことを試みるが、高速割り込み処理を可能とするためには、レジスタバンク10内に未使用のレジスタファイルが存在することが条件となる。そこで、前記S7でレジスタバンクに未使用のレジスタファイルがあるか否かをチェックするのである。

【0025】図2の割り込み制御部内のN進アップダウンカウンタ23の出力がNでない場合（即ちN未満の場合）には、N組あるレジスタファイルのうち未使用のものが存在することを示している。N進アップダウンカウンタ23の計数値はその時点で使用中のレジスタファイル番号（1とNとの間の数）を示している。このN進アップダウンカウンタ23の初期値は1であり、高速割り込みを受け付ける毎に1つつカウントアップする。カ

ウントアップする条件は、高速割り込み要求が上記S 5及びS 6の判別動作により受け付けられたこと、およびS 7の判別によりN進アップダウンカウンタ2 3の計数値がN未満であるとの判別結果の両方が満たされることである。また、このカウンタがカウントダウンする条件は、割り込みリターン命令が実行され、そのときのステータスレジスタ7の該当チェンジフラグの値が“1”であることの両方が満たされることである。

【0026】図5のS 5及びS 6の判別結果が共にYESで、さらにS 7において、N進アップダウンカウンタ2 3の計数値がNでない、つまり、レジスタバンク10内に未使用のレジスタファイルが存在していると判別された場合、高速割り込み処理を行なうことが出来る条件が全て満足されたことになる。図2のANDゲート28は、N進アップダウンカウンタ2 3から出力され、その計数値がNのときは“1”で、Nでないときは“0”である論理信号を、丸印の付加された一方の入力端で論理反転し、この反転信号とモードレジスタ22から出力されその他方の入力端へ供給される該当高速割り込みイネーブル信号との論理積信号であるアペイラブル信号（レジスタバンク10内のファイルレジスタが利用可能であり、S 7の判別結果のYESに相当する）を出力する。ANDゲート28の出力するアペイラブル信号は、N進アップダウンカウンタ2 3には、アップイネーブル機能信号として、シーケンサ24にはロールアップ機能信号として、ベクタ生成部25へは高速ベクタ機能信号としてそれぞれ供給される。

【0027】割り込み制御部9は図5のS 8でレジスタバンク10内のレジスタファイルの切り替えを行う。即ち割り込み前に使用していたレジスタファイルから、割り込み処理で使用する新規のレジスタファイルに切り替える。このレジスタファイルの切り替えは、N進アップダウンカウンタ2 3を+1カウントアップして、その出力であるレジスタファイル選択信号の値を1つ増加することで行なわれる。このレジスタファイル選択信号は、レジスタファイル選択部11に供給され、レジスタファイル選択部11は直ちに選択信号に従った選択動作を行う。従ってレジスタファイル選択信号の値が変わると、直ちに計算機が使用するレジスタファイルが更新されることになる。

【0028】前記レジスタファイルが更新されると、同時に割り込み制御部内のシーケンサ24はS 9において、それまで不使用状態であった該当ファイルレジスタが、更新により使用状態になったことを示すフラグ情報（この例ではチェンジフラグと呼び、不使用では“0”、使用中では“1”となる情報）として“1”の値のチェンジフラグをステータスレジスタ7に与え、ステータスレジスタ7内の該当ファイルレジスタのチェンジフラグ情報を“1”にセットする。その結果、ステータスレジスタ7には、高速割り込み処理としてレジスタ

ファイルを更新したという状態情報が保存される。このチェンジフラグは後述する割り込みリターン命令の実行時に参照される。シーケンサ24がチェンジフラグを“1”にする条件は、N進アップダウンカウンタ2 3のカウントアップ条件と同様で、高速割り込み要求が受け付けられたこと、および、N進アップダウンカウンタ2 3の計数値がNでないことの両方が成立することである。

【0029】最後に、割り込み制御部内のベクタ生成部25は、S 10で、選択された高速割り込み要因に対応する高速割り込みベクタを発生し、この割り込みベクタを図1の分岐制御部8に供給する。ベクタ生成部5において、高速割り込みベクタ（高速割り込みプログラムを選択するデータの意）は、ANDゲート28から供給される高速ベクタ機能信号（源名称はアペイラブル信号）の値が“1”のとき生成される。そしてこの生成されるデータ値は、受け付けられた割り込み要因毎に異なるデータ値となり、受け付け割り込み信号入力値により決定されるものである。そして分岐制御部8は与えられた割り込みベクタから分岐アドレスを生成し、高速割り込み処理ルーチンに分岐する。

【0030】図4は図1のレジスタバンク10内のレジスタファイル1～N（図ではRF 1～RF nと記す）の使用法を説明する図である。図4の左側は、RF 1～RF nの使用状態を示すステータスレジスタ7内のチェンジフラグデータが“110…0”の状態であり、これはRF 1とRF 2が既に使用中であり、RF 3～RF nが未使用であることを示す。この左側の状態で、高速割り込み処理が開始されると、新たにRF 3が選択され、図4の右側に示されるように、ステータスレジスタ7内のチェンジフラグデータは、RF 3も使用中であることを示す“1110…0”の状態になる。そして高速割り込み処理から復帰時に、図の右側に示すように、割り込み処理前の使用状態に戻り、ステータスレジスタ7内のチェンジフラグデータは“110…0”に変更されることを示している。

【0031】割り込みが成立した場合で、上記の高速割り込み処理以外の場合は全て通常割り込み処理を行なう。割り込み要因が高速割り込み要因であったとしても、レジスタバンク10内にあるレジスタファイルが全て使用中の場合にも通常割り込み処理を行う。即ち図5のS 5、S 6、又はS 7のいずれか1つの判別結果がNOの場合には、通常割り込み処理を行うことになる。この通常割り込み処理は、従来技術による割り込み処理装置における動作と同様である。すなわち、プログラムカウンタ1の内容、および、ステータスレジスタ7の内容を使用中のレジスタファイルの特定レジスタに退避した後、図2のベクタ生成部25から割り込み要因毎の通常割り込みベクタを分岐制御部8に与える。割り込み要因毎のベクタは、高速割り込みベクタと同様に、受け付け



割り込み信号入力から決定される。この場合、ANDゲート28から供給される高速ベクタ機能信号の値は

“0”である。これにより、分岐制御部8は図5のS11で、与えられた割り込みベクタから分岐アドレスを生成し、通常割り込み処理ルーチンに分岐する。通常割り込み処理の場合には、レジスタファイルの切り替えは行なわれない。したがって、シーケンサ24からはチェンジフラグの値として“0”がステータスレジスタ7に与えられる。

【0032】図6の(a)は高速割り込みルーチンのフローを、(b)は通常割り込み処理ルーチンのフローを示している。高速割り込み処理では、前記説明のように、レジスタファイルが切り替えられるため、従来のようにレジスタファイルの内容をメモリに退避し、その後これを復帰させる処理は不要である。従って図5の手順により高速割り込み処理が開始されると、直ちにS21で、割り込み処理本体を実行する。通常割り込み処理では、レジスタファイルが切り替わらないため、通常割り込み処理が開始されると、まずS31で、レジスタファイルの内容をメモリ装置2に退避し、S32で、割り込み処理本体を実行し、その終了後のS33で、先にメモリ装置2に退避したレジスタファイルの内容を復帰する処理が必要となる。

【0033】割り込みからの復帰処理は、割り込みリターン命令の実行により行なわれる。割り込みリターン命令は、割り込み処理ルーチンの最後に実行される命令である。割り込みリターン命令は、命令デコード部4からリターン命令制御信号が割り込み制御部9に与えられ処理がはじまる。同時にその時点のステータスレジスタ7内のチェンジフラグの値が割り込み制御部9に与えられる。図6のS21又はS34で、命令デコード部4から割り込みリターン命令が割り込み制御部9に与えられると、図2のN進アップダウンカウンタ23およびシーケンサ24は、リターン命令およびステータスレジスタ7からのチェンジフラグにより割り込み復帰処理を制御する。

【0034】図7のS41において、N進アップダウンカウンタ23及びシーケンサ24は、それぞれ現在使用中のレジスタファイルのチェンジフラグの値が“1”か“0”かを判別する。まず、N進アップダウンカウンタ23はチェンジフラグをチェックし、その値が“1”であった場合には、高速割り込み処理によりレジスタファイルが切り替えられたと認識される。この場合の例は、図4の右側のRF3に対応するステータスレジスタ7内のチェンジフラグデータが“1”の状態として示されている。そしてこの該当チェンジフラグのデータが“1”の場合には、N進アップダウンカウンタ23はS42で、レジスタファイルの切り替えを行う。即ちN進アップダウンカウンタ23には、チェンジ信号がダウンインネーブル機能信号として、またリターン命令がカウントダ

ウン機能信号として入力され、その結果その計数値を-1だけカウントダウンする。この-1のカウントダウンにより、レジスタファイル選択信号は割り込み直前に使用していたレジスタファイルを選択することになる。また該当チェンジフラグのデータが“0”の場合には、上記N進アップダウンカウンタ23のカウントダウン動作は行われない。そしてS44へ移る。

【0035】N進アップダウンカウンタ23の動作と同時に、シーケンサ24もS41で、チェンジフラグの内容をチェックし、その値が“1”の場合には、S43でチェンジフラグのリセットを行う。これは上記N進アップダウンカウンタ23のカウントダウン処理により元のレジスタファイルが選択されるようになるため、チェンジフラグの値を“0”にしてステータスレジスタ7に与える必要があるからである。このためシーケンサ24には、チェンジ信号がロールバック機能信号として、またリターン命令がリカバー機能信号として入力され、その結果“0”のチェンジフラグを出力する。この場合の例は、図4の右側のRF3に対応するチェンジフラグが“1”から“0”に変更されるものとして示される。このステータスレジスタ7内のフラグデータの変更により、レジスタファイルは割り込み直前に使用していたものに巻き戻されたという状態を示す。

【0036】またシーケンサ24がS41で、チェンジフラグの内容をチェックし、チェンジフラグの値が“0”の場合には、通常割り込み処理が行なわれレジスタファイルの切り替えはされていなかったと認識されるため、上記フラグデータの変更処理は行なわれない。そしてS44へ移る。ここまでは、割り込み復帰処理における、高速割り込み処理からの復帰と、通常割り込み処理からの復帰との処理の違いである。以下は、両者の場合においても同一の動作となる。

【0037】割り込み処理制御部9は図7のS44で、ステータスレジスタ7に復帰信号を与え、レジスタファイルの特定レジスタに退避されている割り込み時点のステータスレジスタ7の内容を内部バスを介して読みだしステータスレジスタ7に復帰する。次に割り込み処理部9はS45で、復帰信号をプログラムカウンタ1に与え、レジスタファイルの特定レジスタに退避されている割り込まれた命令アドレスを読みだし、プログラムカウンタ1に復帰する。プログラムカウンタ1からは復帰後の最初の命令アドレスがメモリ装置2に与えられ、割り込みにより中断されたプログラムの実行が再開される。

【0038】なお図1のレジスタファイル選択部11及び図2の割り込み制御部9内の各回路は、高速論理素子により構成されているので、高速割り込み処理が可能な場合に行うレジスタファイルの更新及び復帰のための切り替え所要時間はきわめて短時間であり、従来のレジスタファイルの内容のメモリ装置への退避及び復帰処理は不要となったので、割り込み処理の事前及び事後処理の



所要時間が従来よりも大幅に低減された。また本発明の高速割り込み処理では、更新されたレジスタファイルを使用するので、割り込み処理において使用可能なレジスタ数は、非割り込み処理の場合と同数であり、従来技術における限定された数のレジスタを使用する場合に生じる割り込み処理プログラム記述上の制限が全くない。

【0039】さらに、本発明では、割り込み処理のためのレジスタファイルの使用状況をハードウェアにてチェックし、レジスタファイル切り替えを制御していること、また、レジスタファイルを切り替えた場合とそうでない場合には、別々の割り込みベクタを発生する機構をもたせている。このため、従来方式による割り込み処理のための特別なレジスタセットへの切り替えを行なうために必要な、毎割り込み受け付け時にソフトウェアにより特別なレジスタセットを使用可能か否かの判定を行なうことは不要となり、この判定処理による性能低下の問題はすべて回避できる。

【0040】

【発明の効果】以上のように本発明に係る割り込み処理装置は、複数のレジスタファイルから構成されるレジスタバンクと、プロセッサに係る状態情報を保持する状態レジスタと、複数の通常割り込み信号及び複数の高速割り込み信号を入力し、それぞれの割り込み処理プログラムへの分岐及び復帰を制御する割り込み制御手段とを備えており、前記割り込み制御手段は通常割り込み制御手段と高速割り込み制御手段とを有する。通常割り込み制御手段は、前記通常割り込み信号が入力された場合に、その時点における前記状態レジスタの保持情報及びプログラム戻りアドレスを退避保存し、該当通常割り込み信号により一義的に与えられる割り込みベクタから所定の割り込み処理プログラムに分岐させ、この分岐した割り込み処理プログラムの実行終了後に、前記退避保存した状態レジスタの保持情報を状態レジスタに復帰させると共に退避保存したプログラム戻りアドレスから分岐前に実行中のプログラムに復帰させ、高速割り込み制御手段は、前記高速割り込み信号が入力された場合に、その時点における前記状態レジスタの保持情報及びプログラム戻りアドレスを退避保存し、次に前記レジスタバンク内に使用中でないレジスタファイルの有無を判別して有る場合には、その時点で使用中のレジスタファイルの代りにレジスタバンク内の使用中でないレジスタファイルを割り付け、該当高速割り込み信号により一義的に与えられる割り込みベクタから所定の割り込み処理プログラムに分岐させ、この分岐した割り込み処理プログラムの実行終了後に、分岐前に使用していたレジスタファイルに戻し、前記退避保存した状態レジスタの保持情報を状態レジスタに復帰させると共に退避保存したプログラム戻りアドレスから分岐前に実行中のプログラムに復帰させ、また前記レジスタバンク内に使用中でないレジスタファイルの有無を判別して無い場合には、レジスタファ

イルの新規割り付けは行わずに、前記通常割り込み制御手段による制御を行わせるための別の割り込みベクタで指定される割り込み処理プログラムに分岐させるようにしたので、高速割り込み処理が可能な場合には、その時点で使用中のレジスタファイルをハードウェアにより高速で切り替え高速割り込み処理プログラムに分岐し、この分岐したプログラムの実行終了後に、再び分岐前に使用していたレジスタファイルに高速で戻して分岐前のプログラムに復帰可能で、従来方式のようにレジスタファイルの内容のメモリ装置への退避処理及び復帰処理が不要となり、従来よりも割り込み処理の事前処理及び事後処理時間が大幅に低減され、割り込み処理性能が向上する。また本発明では、高速割り込み処理時にレジスタファイルそのものを切り替えるので、レジスタファイル内に含まれるレジスタの数は非割り込み処理である通常処理と同数であり、その結果、従来方式におけるレジスタファイルの一部のみを退避及び復帰させ、限られた数のレジスタを使用して割り込み処理を行う場合に生じる割り込み処理プログラム記述上の制限が全くない。

【図面の簡単な説明】

【図1】本発明の実施例による割り込み処理装置の構成を示す図である。

【図2】図1の割り込み処理部9の具体的な構成例を示す図である。

【図3】図2のモードレジスタ22の構成を示す図である。

【図4】図1のレジスタファイル1～Nの使用方法を説明する図である。

【図5】本発明の実施例による割り込み開始処理の流れ図である。

【図6】本発明の実施例による高速及び通常割り込み処理の流れ図である。

【図7】本発明の実施例による割り込み復帰処理の流れ図である。

【図8】従来の計算機の割り込み処理装置の構成を示す図である。

【図9】図8の装置の割り込み処理の開始、実行及び復帰順序を示す流れ図である。

【符号の説明】

- 1 プログラムカウンタ
- 2 メモリ装置
- 3 命令レジスタ
- 4 命令デコード部
- 5 命令実行部
- 7 ステータスレジスタ
- 8 分岐制御部
- 9 割り込み制御部
- 10 レジスタバンク
- 11 レジスタファイル選択部

【図 3】

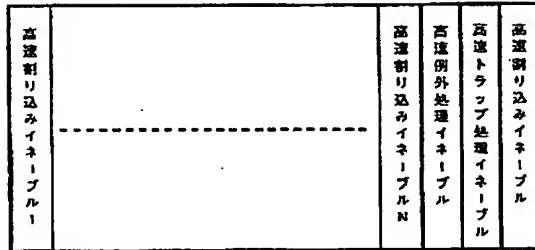


図2のモードレジスタ22の構成を示す図

【図 4】

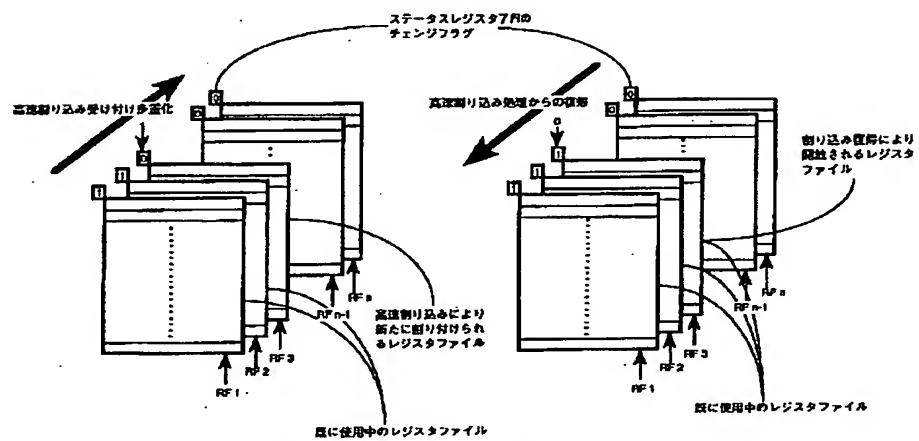
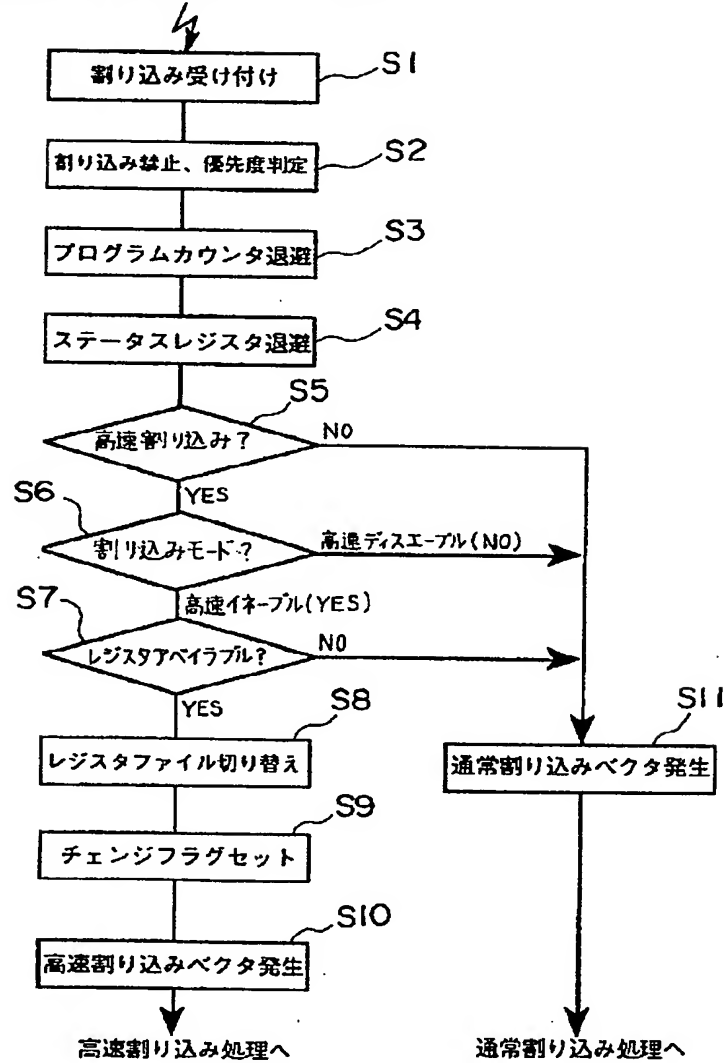


図1のレジスタファイル1～Nの使用方法を説明する図

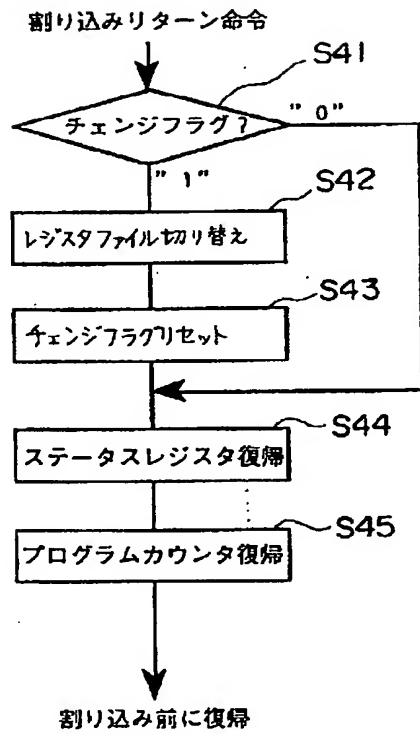
【図5】

例外発生、割り込み入力、トラップ命令



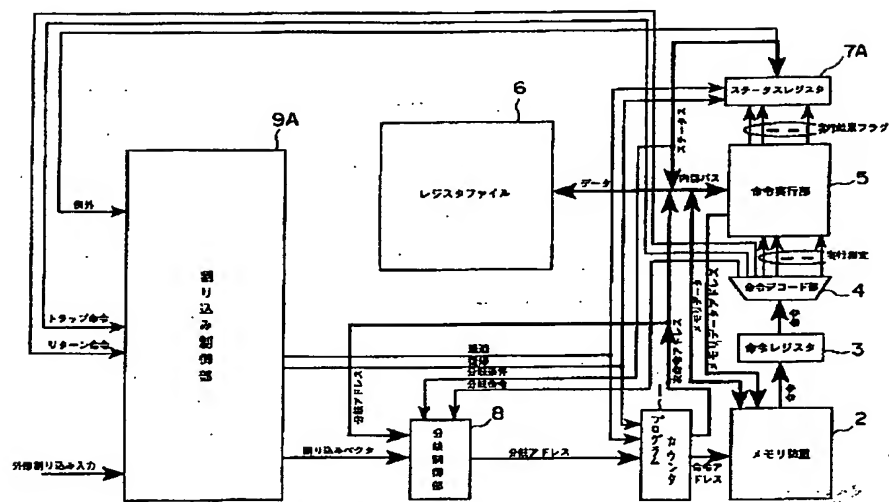
本発明の実施例による割り込み開始処理の流れ図

【図7】



本発明の実施例による割り込み復帰処理の流れ図

【図8】



従来の計算機の割り込み処理装置の構成を示す図

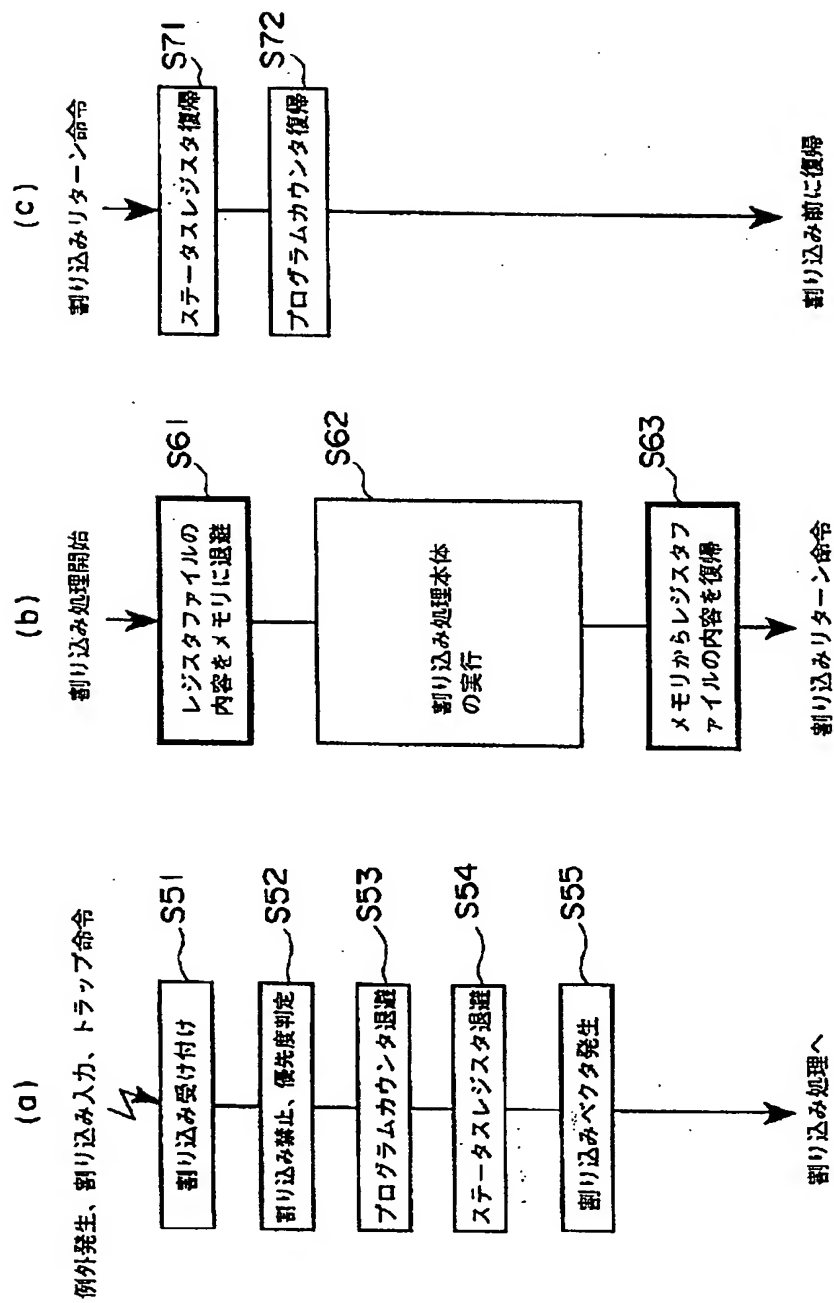


図8の装置の割り込み処理の開始、実行及び復帰順序を示す流れ図

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-212083

(43)Date of publication of application : 20.08.1996

(51)Int.Cl.

G06F 9/46  
G06F 9/46

(21)Application number : 07-019383

(71)Applicant : OKI ELECTRIC IND CO LTD

(22)Date of filing : 07.02.1995

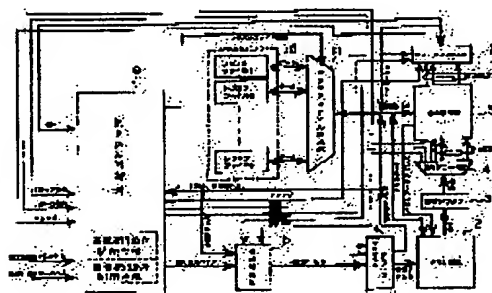
(72)Inventor : KAWAI ATSUSHI

## (54) INTERRUPTION PROCESSOR

### (57)Abstract:

**PURPOSE:** To provide an interruption processor which can shorten the time needed for the saving/return processing of the register contents in an interruption processing mode and also can improve the interruption processing performance.

**CONSTITUTION:** An interruption processor is provided with a register bank 10 which consists of register files 1 to N, a status register 7 which holds the state information on the processor, and an interruption control part 9 which inputs plural normal interruption signals and plural fast interruption signals and controls the branch and return of these signals to every interruption program. The part 9 has a normal/fast interruption control means. This control means returns the information held by a state register when an interruption signal is inputted together with the saving information obtained after the branch/ interruption processing is finished by the saving of a program return address and by the generation of an interruption vector, branches the interruption signal by the interruption vector that is generated after allocation of a new register file as long as the fast interruption processing is possible, and then returns the saving information to the register file that is used before branch of the interruption signal after the interruption processing is finished.



## LEGAL STATUS

[Date of request for examination] 27.07.2001

[Date of sending the examiner's decision of rejection] 18.01.2005

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's  
decision of rejection]

[Date of extinction of right]



## NOTICES \*

IPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.

2. \*\*\*\* shows the word which can not be translated.

3. In the drawings, any words are not translated.

## CLAIMS

## Claim(s)

[Claim 1] The register bank which consists of two or more register files, and the status register holding the status information concerning a processor, It is the interruption processor equipped with an interrupt control means to input two or more usual interrupt signal and two or more high-speed interrupt signals, and to control branching and the return to each interruption processing program. Said interrupt control means when [ said ] an interrupt signal is usually inputted Evacuation preservation of the maintenance information and the program return address of said status register at the time is carried out. A predetermined interruption processing program is branched from the interrupt vector table uniquely given by the applicable usual interrupt signal. The usual interrupt control means returned to the program under activation from the program return address which carried out evacuation preservation while returning the maintenance information on said status register which carried out evacuation preservation to the status register after activation termination of this branched interruption processing program before branching. When said high-speed interrupt signal is inputted, evacuation preservation of the maintenance information and the program return address of said status register at the time is carried out. next, in distinguishing the existence of the register file which is not used in said register bank and being The register file which is not used in a register bank instead of a register file in use at the time is assigned. A predetermined interruption processing program is branched from the interrupt vector table uniquely given by the applicable high-speed interrupt signal. It returns to the register file which was being used before branching after activation termination of this branched interruption processing program. It is made to return to the program under activation from the program return address which carried out evacuation preservation while returning the maintenance information on said status register which carried out evacuation preservation to the status register before branching. moreover, distinguish the existence of the register file which is not used in said register bank, and when there is nothing It is the interruption processor characterized by having a high-speed interrupt control means to branch the interruption processing program specified by another interrupt vector table for making said control usually according to an interrupt control means perform, without performing new allotment of a register file.

[Claim 2] The mode register constituted by the various interruption enabling flags with which said interrupt control means contains two or more high-speed interruption enabling flags of each, The priority judging means which inputs the maintenance information on said mode register into said two or more high-speed interrupt signals and two or more interruption factor signals which usually include an interrupt signal, and a list, and carries out the selection output of the signal of the highest priority of two or more interruption factor signals, The addition-and-subtraction counter which carries out counting of the number of register files in use [ in said REJITA bank ], and outputs a register file selection signal, A vector generation means to generate the interrupt vector table for branching processing based on the output of the output of said priority judging means, the maintenance information on a mode register, and an addition-and-subtraction counter, The interruption processor according to claim 1 characterized by having a sequence control means to output each control signal of said evacuation preservation, a return, and modification of the maintenance information on a status register, based on the output of the maintenance information on said status register, and a priority judging means, and return instruction.

**\* NOTICES \***

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

---

**DETAILED DESCRIPTION**

---

**[Detailed Description of the Invention]****[0001]**

**[Industrial Application]** This invention relates to the interruption processor equipped with the high-speed interruption processing means.

**[0002]**

**[Description of the Prior Art]** Drawing 8 is drawing showing the configuration of the interruption processor of the conventional computer. drawing 8 -- setting -- 1 -- a program counter and 2 -- a memory apparatus and 3 -- for the instruction-execution section and 6, a register file and 7A are [ an instruction register and 4 / the instruction decoding section and 5 / a branching control section and 9A of the conventional status register and 8.] the conventional interrupt control sections. Drawing 9 is the flow chart showing initiation of interruption processing of the equipment of drawing 8, activation, and return sequence.

**[0003]** Actuation of drawing 8 is explained with reference to drawing 9. In drawing 8, a program counter 1 is the part which generates the instruction address which should be performed next, and the instruction address is given to a memory apparatus 2 from here. A memory apparatus 2 reads the instruction code which should be performed from the given instruction address, and stores it in an instruction register 3. The instruction code stored in the instruction register 3 is changed into the control signal which is decoded in the instruction decoding section 4 (decode), and is given to the instruction-execution section 5. The instruction-execution section 5 processes according to the control signal for the given instruction-execution assignment. At this time, the operand data (data set as the object of an operation) used as the object processed is read from a register file 6 through an internal bus. Moreover, processing result data are similarly stored in the assignment register in a register file 6 from the instruction-execution section 5 through an internal bus. The register assignment in a register file 6 is expressed in instruction code. The activation result flag which can come, simultaneously reflects an instruction-execution result is stored in status register 7A. The usual instruction execution is carried out in this way, and is performed.

**[0004]** When the instruction which should be executed is a memory access instruction, memory address count is performed in the instruction-execution section 5. When a memory access instruction is a memory readout, memory access is performed by giving the calculated memory address to a memory apparatus 2. The data read from memory are stored in the register specified by the instruction code in a register file 6 through an internal bus. Moreover, when a memory access instruction is memory writing, memory access is performed by giving the memory write-in data read from the memory address calculated in the instruction-execution section 5, and the register specified by the instruction code in a register file 6 to a memory apparatus 2 through an internal bus, and data are written in a predetermined memory address. The activation result flag which reflects an instruction-execution result in this and coincidence is stored in a status register 7. The usual instruction execution is carried out in this way, and is performed.

**[0005]** When the instruction which should be executed is branch instruction, the branching place instruction address is calculated in the instruction-execution section 5. The calculated branch address is given to the branching control section 8. From the instruction decoding section 4, the decision-instruction signal for branching control is given to coincidence, and branch condition is given

to the branching control section 8 from status register 7A, respectively. In the branching control section 8, it judges whether branch condition is satisfied from these input signals, and when branch condition is satisfied, the branch address which should be performed next is given to a program counter 1.

[0006] When the instruction which executed immediately before when the instruction which should be executed is a trap instruction (conditional [ to the specified address / which is not planned ] instruction with which the location of the origin over which the jump transfer has been carried out, hardware started automatically, and it has jumped is recorded) generates exception condition, such as overflow, an underflow, and division by zero, it means an interrupt request generating in one when an external interruption input is given of cases. In this case, at that time, the program execution under activation is interrupted temporarily and interruption processing corresponding to each interruption factor is performed.

[0007] In order to realize this interruption processing, in the step 51 of (a) of drawing 9 (it is only described as S below), first, said exception condition, a trap instruction, and an external interruption input are inputted into interrupt control section 9A, and are received. Interrupt control section 9A is S52, and removes the factor to which interruption is first forbidden among these interruption factors. And what has the highest priority is chosen from the factors which remained. When interruption is materialized as a result of the aforementioned selection, interrupt control section 9A is S53 first, gives an evacuation signal to a program counter 1, and evacuates the interrupted instruction address to the specific register of a register file 6 through an internal bus.

[0008] Next, by S54, an evacuation signal is given to status register 7A, and the contents of status register 7A at the time of interrupting are similarly evacuated to the specific register of a register file 6 through an internal bus. Then, interrupt control section 9A is S55, and it interrupts and it outputs the selected interrupt vector table (the method which distributes an interrupt signal to the program which corresponds with that value, respectively is called vector interruption, and the interrupt vector table in this case is the mind of the data which choose a corresponding program.) corresponding to a factor to the branching control section 8. In the branching control section 8, the start address of an actual interrupt handler is calculated from the given interrupt vector table, and it outputs to a program counter 1. It performs after this by reading the instruction which should be executed from a memory apparatus 2 with the instruction address given from a program counter 1 like the usual instruction execution.

[0009] An instruction execution shifts to an interrupt handler through the above-mentioned sequence. In an interrupt handler, in S61 [ of drawing 9 ] of (b), interrupt control section 9A interrupts first, and usually evacuates the contents of the register file 6 at the time to a memory apparatus 2. This is performed as an instruction in an interruption processing program. Even if this uses a register file 6 by interruption processing, the contents of the register file 6 before interruption will be saved. And original interruption processing is performed in S62. When interruption processing is completed, interrupt control section 9A returns the contents of the register file 6 in the time of interrupting at which it has evacuated to the memory apparatus 2 contrary to S61 of previously to a register file 6 in S63. This processing is also performed as an instruction in an interruption processing program. And it interrupts in order to return from interruption processing finally, and a return instruction is executed.

[0010] A return instruction control signal is given to interrupt control section 9A from the instruction decoding section 4, and, as for an interruption return instruction, processing starts. In S71 [ of drawing 9 ] of (c), interruption processing control-section 9A gives a return signal first to status register 7A, reads the contents of status register 7A at the interruption time evacuated to the specific register of a register file 6 through an internal bus, and returns to status register 7A. Next, in S72, a return signal is given to a program counter 1, the interrupted instruction address which is evacuated to the specific register of a register file 6 is read, and it returns to a program counter 1. From a program counter 1, the instruction address of the beginning after a return is given to a memory apparatus 2, and the program execution interrupted by interruption is resumed.

[0011]

[Problem(s) to be Solved by the Invention] However, in the above interruption processors of a calculating machine, since a register file had only a lot, the contents of the register file which

evacuated the contents of the register file to the memory apparatus at the time of initiation of interruption processing, and evacuated at the time of termination of interruption processing needed to be returned from the memory apparatus. There are two kinds, the case where evacuation of the contents of this register file and return processing are performed by the instruction execution, and when being carried out by hardware, of cases. Since the time amount which corresponds for this processing also in any execution time of dozens instructions or case was spent on evacuation and each return, the interruption response time and the interruption processing time became long, and there was a trouble of having brought about degradation. When high-speed interruption response processing is required especially, interruption processing may be performed using a number of registers to which it evacuated and returned and some contents of the above-mentioned register file were restricted. However, evacuation of a register and the time amount of restoring are required also in this case. Moreover, by programming which uses a reduced number of registers, a big limit will be given to interruption processing program description, and it becomes remarkably difficult to consider as interruption processing programming with a high level language and the high interruption processing program of portability. Degradation may be brought about on the contrary for the program which used only a small number of register.

[0012] Moreover, in order to avoid the above-mentioned problem, there is also a computer which is preparing the register set special for interruption processing etc. The problem by this performing evacuation of a register file and a return is mitigable. However, the number of groups of the special register set prepared is limited, and excluding evacuation of a register file and a return in all situations that an interrupt generates in multiplex is not guaranteed. Therefore, in such a calculating machine, it judged with software whether it would be usable in a special register set at the time of a \*\*\*\*\* receptionist, and only when possible, evacuation of a register file and a return are excluded. Even if it was able to perform interruption processing using the register set special at the time of interruption generating temporary almost, the usable judgment had to be performed by software in it, and the trouble of having brought about degradation also had a duration for it.

[0013]

[Means for Solving the Problem] The register bank where the interruption processor concerning this invention consists of two or more register files. Two or more usual interrupt signals and two or more high-speed interrupt signals are inputted as the status register holding the status information concerning a processor. It is the interruption processor equipped with an interrupt control means to control branching and the return to each interruption processing program. Said interrupt control means When [ said ] an interrupt signal is usually inputted, evacuation preservation of the maintenance information and the program return address of said status register at the time is carried out. A predetermined interruption processing program is branched from the interrupt vector table uniquely given by the applicable usual interrupt signal. The usual interrupt control means returned to the program under activation from the program return address which carried out evacuation preservation while returning the maintenance information on said status register which carried out evacuation preservation to the status register after activation termination of this branched interruption processing program before branching. When said high-speed interrupt signal is inputted, evacuation preservation of the maintenance information and the program return address of said status register at the time is carried out. next, in distinguishing the existence of the register file which is not used in said register bank and being The register file which is not used in a register bank instead of a register file in use at the time is assigned. A predetermined interruption processing program is branched from the interrupt vector table uniquely given by the applicable high-speed interrupt signal. It returns to the register file which was being used before branching after activation termination of this branched interruption processing program. It is made to return to the program under activation from the program return address which carried out evacuation preservation while returning the maintenance information on said status register which carried out evacuation preservation to the status register before branching. moreover, distinguish the existence of the register file which is not used in said register bank, and when there is nothing It has a high-speed interrupt control means to branch the interruption processing program specified by another interrupt vector table for making said control usually according to an interrupt control means perform, without

performing new allotment of a register file.

[0014]

[Function] The interruption processor concerning this invention inputted two or more usual interrupt signals and two or more high-speed interrupt signals as the register bank which consists of two or more register files, and the status register holding the status information concerning a processor, it has an interrupt control means to control branching and the return to each interruption processing program, and said interrupt control means usually has an interrupt control means and a high-speed interrupt control means. An interrupt control means usually, when [ said ] an interrupt signal is usually inputted Evacuation preservation of the maintenance information and the program return address of said status register at the time is carried out. A predetermined interruption processing program is branched from the interrupt vector table uniquely given by the applicable usual interrupt signal. While returning the maintenance information on said status register which carried out evacuation preservation to a status register after activation termination of this branched interruption processing program, it is made to return to the program under activation from the program return address which carried out evacuation preservation before branching. When said high-speed interrupt signal is inputted, a high-speed interrupt control means Evacuation preservation of the maintenance information and the program return address of said status register at the time is carried out. next, in distinguishing the existence of the register file which is not used in said register bank and being The register file which is not used in a register bank instead of a register file in use at the time is assigned. A predetermined interruption processing program is branched from the interrupt vector table uniquely given by the applicable high-speed interrupt signal. It returns to the register file which was being used before branching after activation termination of this branched interruption processing program. It is made to return to the program under activation from the program return address which carried out evacuation preservation while returning the maintenance information on said status register which carried out evacuation preservation to the status register before branching. moreover, distinguish the existence of the register file which is not used in said register bank, and when there is nothing The interruption processing program specified by another interrupt vector table for making said control usually according to an interrupt control means perform is branched without performing new allotment of a register file. In this invention therefore, when high-speed interruption processing is possible Change a register file in use by hardware at high speed at the time, and it branches to a high-speed interruption processing program. Since it returns to the register file which was being again used before branching at high speed and the program before branching is returned after this branched program execution termination The evacuation processing to the memory apparatus of the contents of the register file and return processing become unnecessary like the conventional method, rather than before, prior processing and corrective action time amount of interruption processing are reduced sharply, and the interruption processing engine performance improves. Moreover, since the register file itself is changed in this invention at the time of high-speed interruption processing, the number of the registers contained in a register file does not have the limit on the interruption processing program description produced when performing interruption processing using a number of registers which a part of register file [ in / it is usually processing and the same number, consequently / the conventional method ] which is processing in which it does not interrupt was evacuated and returned, and were restricted.

[0015]

[Example] Drawing 1 is drawing showing the configuration of the interruption processor by the example of this invention. In drawing 1, 1-5, it reaches, 8 is the same as that of drawing 8, and 7 is a status register by this invention. Generally, although a status register holds the status information (for example, instruction-execution result flag etc.) concerning a processor, the status register 7 of this invention also includes the flag information (in this example, drawing 4 explains a change flag and a call detail) which shows whether two or more register files of each in a register bank other than the status information concerning this general processor are using it. 9 is the interrupt control section by this invention, and is an interrupt control means to input two or more high-speed interrupt signals and two or more usual interrupt signals, and to control branching and the return to each interruption processing. Therefore, the interrupt control section 9 usually includes an interrupt control means with

a high-speed interrupt control means functionally. 10 is a register bank containing two or more register files 1 of N individual - N, and said status register 7 to reading of the flag information which each register file is using it or shows \*\*\*\* has become possible. 11 is the register file selection section, chooses one register file specified by the register file selection signal supplied from the interrupt control section 9, and makes connection with an internal bus.

[0016] Actuation of drawing 1 is explained. In drawing 1, a program counter 1 is the part which generates the instruction address which should be performed next, and the instruction address is given to a memory apparatus 2 from here. A memory apparatus 2 reads the instruction code which should be performed from the given instruction address, and stores it in an instruction register 3. The instruction code stored in the instruction register 3 is changed into the control signal which is decoded in an operator part and given to the instruction-execution section 5. The instruction-execution section 5 processes according to the control signal for the given instruction-execution assignment. At this time, the operand data used as the object processed is read from the register with which the specific register file in the register file 1 - N in a register bank 10 is specified through an internal bus. This specific register file is chosen by the register file selection signal given from the interrupt control section 9, and is connected to an internal bus. The register assignment in this specific register file is expressed in instruction code. Moreover, processing result data are stored in the register with which it is similarly specified in the register file of above specification from the instruction-execution section 5 through an internal bus. The activation result flag which can come, simultaneously reflects an instruction-execution result is stored in a status register 7. The usual instruction execution is carried out in this way, and is performed.

[0017] When the instruction which should be executed is a memory access instruction, memory address count is performed in the instruction-execution section 5. When a memory access instruction is a memory readout, memory access is performed by giving the calculated memory address to a memory apparatus 2. The data read from memory are stored in the register specified by the instruction code in a register file through an internal bus. Moreover, when a memory access instruction is memory writing, memory access is performed by giving the memory write-in data read from the memory address calculated in the instruction-execution section 5, and the register specified by the instruction code in a register file to a memory apparatus 2 through an internal bus, and data are written in a predetermined memory address.

[0018] When the instruction which should be executed is branch instruction, the branching place instruction address is calculated in the instruction-execution section 5. The calculated branch address is given to the branching control section 8. From the instruction decoding section 4, the decision-instruction signal for branching control is given to coincidence, and branch condition is given to the branching control section 8 from a status register 7, respectively. In the branching control section 8, it judges whether branch condition is satisfied from these input signals, and when branch condition is satisfied, the branch address which should be performed next is given to a program counter 1. When the instruction executed immediately before when the instruction which should be executed was a trap instruction generates exception condition, such as overflow, an underflow, and division by zero, or when a high-speed external interruption input or a usual external interruption input is given, it is recognized as the interrupt request having occurred in any case. In these cases, at the time, the program execution under activation is interrupted temporarily and interruption processing corresponding to each interruption factor is performed.

[0019] Drawing 5 is the flow chart of the interruption initiation processing by the example of this invention, and is the flow chart of the high speed according [ drawing 6 ] to the example of this invention, and usual interruption processing, drawing 2 is drawing showing the concrete example of a configuration of the interrupt control section 9 of drawing 1, and drawing 3 is drawing showing the configuration of the mode register 22 of drawing 2, and it is [ it is drawing where drawing 4 explains the operation of the register file 1 of drawing 1 - N, and / drawing 7 interrupts according to the example of this invention, and ] the flow chart of return processing. In drawing 2, it is the AND gate where 21 outputs a priority judging block and 22 outputs the AND of a mode register, and the reversal signal of the signal of an input edge and the signal of the input edge of another side with which, as for the vector generation section, and 26 and 27, amplifier was added, and, as for a sequencer and 25, the



round mark was added [ 23 ] for an N-ary updown counter and 24, as for 28.

[0020] With reference to drawing 1 - drawing 4 , the interruption processing by the example of this invention is explained according to the flow chart of drawing 5 - drawing 7 . First, in S1 of drawing 5 , exception condition, a trap instruction, a high-speed external interruption input, and a usual external interruption input are inputted into the interrupt control section 9 of drawing 1 , and the interruption processing by the example of this invention is received. Nine is the interrupt control section S2, it interrupts to these interruption factor inputs, and a priority is judged to the interruption input in which two or more receptionists are possible to be the judgment of receptionist propriety. And the above-mentioned judgment is performed by the priority judging block 21 of drawing 2 .

[0021] In the priority judging block 21, a check with a respectively possible interruption receptionist is performed to each inputted interruption factor. This is tested by comparison by the contents of the interruption mask register in the priority judging block 21, and the level register which can be interrupted, and is judged. That is, the interruption mask flag for each interruption factor of every beforehand set up by the program is "0", and only when higher than the level in the time of an interrupt request being given which can be interrupted, the interruption factor concerned is received. A receptionist becomes possible only when higher than the factor over the interruption processing which is processing the level which can be interrupted at the time when a calculating machine is already interruption [ of a certain level ] processing the level which can be interrupted serially reflecting the level of processing of a calculating machine at a certain time. Furthermore, when the interrupt request in which two or more interruption receptionists are possible is given to coincidence, what has the highest priority is chosen. Generally interrupt priority is beforehand defined by hardware and is incorporated as wired logic (fixed priority logic) in the priority judging block 21.

[0022] When interruption is materialized as a result of selection by the aforementioned priority judging, the interrupt control section 9 is S3 first, gives an evacuation signal to a program counter 1, and evacuates the interrupted instruction address to the specific register of a register file in use through an internal bus at the time. Next, an evacuation signal is given to a status register 7 by S4, and the contents of the status register 7 at the time of interrupting are similarly evacuated to the specific register of a register file in use through an internal bus at the time. After this, in the example of this invention, the thing which was chosen by the above-mentioned actuation and which interrupts and carries out high-speed interruption processing to a factor so much, and the thing which usually performs interruption processing are judged, and respectively different actuation is performed. In either a trap instruction, exception condition or a high-speed interruption input the interruption factor chosen with the priority judging block 21 performs high-speed interruption processing according to the established state of the mode register 22 of drawing 2 .

[0023] In drawing 3 , the configuration of the mode register 22 of drawing 2 is shown, and a mode register 22 is constituted from each flag of high-speed interruption enabling [ which is shown in drawing 3 ] 1 - N, and high-speed exception-handling enabling, high-speed trap processing enabling, and high-speed interruption enabling. Among the four above-mentioned kinds of enabling flags, high-speed interruption enabling is master enabling, and only when this flag is "1", it becomes effective [ other flags ]. Therefore, when high-speed interruption enabling is "0", high-speed interruption processing is not performed to any interrupt request conditions. When high-speed interruption enabling [ 1 ] - N support the high-speed external interruption input 1 - N, respectively, a high-speed external interruption input is given and the high-speed interruption enabling flag corresponding to this is "1", applicable high-speed interruption processing is performed. Hereafter, when, as for high-speed exception-handling enabling, an exception signal is given, and when, as for high-speed trap processing enabling, a trap instruction is given, that of high-speed interruption enabling [ 1 ] - N, and Hitoshi carries out \*\*\*\*\*.

[0024] The priority judging block 21 distinguishes whether the conditions which reach [ whether it is the inputted factor as which it interrupts and a factor requires high-speed interruption processing, and ], and the set point of a mode register 22 gives are satisfied in S5 and S6 of drawing 5 . And when both the distinction results of S5 and S6 are YES(s), whether nine having the register file which is not used in a register bank 10 in the interrupt control section S7 and a register file distinguish whether it is ABEIRAABURU, and when this distinction result is YES, high-speed interruption processing is



performed. Although it tries to change this from the register file which was being used in the high-speed interruption processing by this invention when it interrupted to a new intact register file, and to perform interruption processing, in order to enable high-speed interruption processing, it becomes conditions that an intact register file exists in a register bank 10. Then, it is confirmed whether an intact register file is in a register bank by said S7.

[0025] When the output of the N-ary updown counter 23 of interrupt control circles of drawing 2 is not N (in namely, the case of under N), it is shown that an intact thing exists among the register files which have N sets. The enumerated data of the N-ary updown counter 23 show the register file number (number between 1 and N) in use at the time. The initial value of this N-ary updown counter 23 is 1, and whenever it receives high-speed interruption, it counts it up one [ at a time ]. The conditions to count up are that the high-speed interrupt request was received by distinction actuation of the above S5 and S6, and that both distinction results that the enumerated data of the N-ary updown counter 23 are under N are fulfilled by distinction of S7. Moreover, the conditions which this counter counts down are that an interruption return instruction is executed and it is fulfilled both that the value of the applicable change flag of the status register 7 at that time is "1."

[0026] Both the distinction results of S5 and S6 of drawing 5 are YES(s), and when the enumerated data of the N-ary updown counter 23 were not N in S7, i.e., the intact register file existed in the register bank 10 and it is distinguished, it means that all the conditions that can perform high-speed interruption processing were satisfied further. The AND gate 28 of drawing 2 is outputted from the N-ary updown counter 23, and it is "1" when the enumerated data are N. The logic signal which is "0" when it is not N was added to while to the round mark, and logic reversal is carried out at an input edge. The available signal which is an AND signal with the applicable high-speed interruption enable signal which is outputted from this reversal signal and mode register 22, and is supplied to the input edge of that another side (the file register in a register bank 10 is available) YES of the distinction result of S7 -- corresponding -- it outputs. the available signal which the AND gate 28 outputs -- the N-ary updown counter 23 is supplied as a high-speed vector ability signal as a roll-up functional signal as a rise enabling functional signal to the vector generation section 25 at a sequencer 24, respectively.

[0027] The interrupt control section 9 changes the register file in a register bank 10 by S8 of drawing 5. That is, it changes from the register file which was being used before interruption to the new register file used by interruption processing. The change of this register file is performed by counting up the N-ary updown counter 23 +one time, and increasing one value of the register file selection signal which is that output. This register file selection signal is supplied to the register file selection section 11, and the register file selection section 11 performs selection actuation which followed the selection signal immediately. Therefore, when the value of a register file selection signal changes, the register file which a calculating machine uses immediately will be updated.

[0028] If said register file is updated, the sequencer 24 of interrupt control circles will be set to coincidence at S9. Flag information which shows that the applicable file register which was a non-busy condition till then would be in the busy condition by updating (in this example, it is called a change flag) In un-using it, in "0" and use, the change flag of the value of "1" is given to a status register 7 as information used as "1", and the change flag information on the applicable file register in a status register 7 is set to "1." Consequently, the status information of having updated the register file as high-speed interruption processing is saved at a status register 7. This change flag is referred to at the time of activation of the interruption return instruction mentioned later. The conditions to which a sequencer 24 sets a change flag to "1" are the same as the count-up conditions of the N-ary updown counter 23, and are that it is materialized that the high-speed interrupt request was received and that the enumerated data of the N-ary updown counter 23 are not [ both ] N.

[0029] The vector generation section 25 of interrupt control circles is S10, generates the high-speed interrupt vector table corresponding to the high-speed selected interruption factor at the last, and supplies this interrupt vector table to it at the branching control section 8 of drawing 1. In the vector generation section 5, a high-speed interrupt vector table (mind of the data which choose a high-speed interruption program) is generated, when the value of the high-speed vector ability signal (a source name is an available signal) supplied from the AND gate 28 is "1." And this data value

generated turns into a received data value which interrupts and is different for every factor, and is determined by the value of a receptionist interrupt signal input. And the branching control section 8 generates a branch address from the given interrupt vector table, and branches to a high-speed interrupt handler.

[0030] Drawing 4 is drawing explaining the usage of the register file 1 in the register bank 10 of drawing 1 - N (it is described as RF1-RFn by a diagram). The change flag data in the status register 7 which the left-hand side of drawing 4 shows the busy condition of RF1-RFn are in the condition of "110-0", RF1 and RF2 are already using this, and it shows that RF3-RFn are intact. If high-speed interruption processing is started in the state of this left-hand side, RF3 is newly chosen, and as shown in the right-hand side of drawing 4, the change flag data in a status register 7 will be in the condition of "1110-0" which shows that RF3 is also under use. And as shown in the right-hand side of drawing from high-speed interruption processing at the time of a return, it is shown that the change flag data in return and a status register 7 are changed into the busy condition before interruption processing by "110-0."

[0031] By the case where interruption is materialized, as for all the cases other than the above-mentioned high-speed interruption processing, interruption processing is usually performed. Even if an interruption factor is a high-speed interruption factor, also when all the register files in a register bank 10 are using it, interruption processing is usually performed. That is, when any one distinction result of S5 and S6 of drawing 5, or S7 is NO, interruption processing will usually be performed. It is the same as that of the actuation in this usually according [ interruption processing ] to conventional technique interruption processor. That is, after evacuating to the specific register of a register file while using the contents of the program counter 1, and the contents of the status register 7, it interrupts from the vector generation section 25 of drawing 2, and the usual interrupt vector table for every factor is given to the branching control section 8. The vector for every interruption factor is determined from a receptionist interrupt signal input like a high-speed interrupt vector table. In this case, the value of the high-speed vector ability signal supplied from the AND gate 28 is "0." Thereby, the branching control section 8 is S11 of drawing 5, generates a branch address from the given interrupt vector table, and usually branches to an interrupt handler. Usually, in interruption processing, the change of a register file is not performed. Therefore, from a sequencer 24, "0" is given to a status register 7 as a value of a change flag.

[0032] (a) of drawing 6 shows the flow of a high-speed interruption routine, and (b) usually shows the flow of an interrupt handler. The processing to which the contents of the register file are evacuated to memory like before, and this is returned after that at high-speed interruption processing since a register file is changed like said explanation is unnecessary. Therefore, shortly after high-speed interruption processing is started by the procedure of drawing 5, an interruption processing body is performed by S21. Usually, since a register file does not change, when interruption processing is usually started, first, the contents of the register file are evacuated to a memory apparatus 2, an interruption processing body is performed by S32, and the processing which returns the contents of the register file which evacuated to the memory apparatus 2 previously by S33 after the termination is needed by S31, with interruption processing.

[0033] Return processing from interruption is performed by activation of an interruption return instruction. An interruption return instruction is an instruction executed by the last of an interrupt handler. A return instruction control signal is given to the interrupt control section 9 from the instruction decoding section 4, and, as for an interruption return instruction, processing starts. The value of the change flag in the status register 7 at the time is given to coincidence at the interrupt control section 9. If it interrupts from the instruction decoding section 4 and a return instruction is given to the interrupt control section 9 by S21 of drawing 6, or S34, the N-ary updown counter 23 and sequencer 24 of drawing 2 will interrupt with a return instruction and the change flag from a status register 7, and will control return processing.

[0034] In S41 of drawing 7, as for the N-ary updown counter 23 and a sequencer 24, the value of the change flag of a register file present in use distinguishes "1" and "0", respectively. First, the N-ary updown counter 23 checks a change flag, and when the value is "1", it is recognized that the register file was changed by high-speed interruption processing. The change flag data in the status register 7

corresponding to RF3 on the right-hand side of drawing 4 in the example in this case are shown as a condition of "1." And when the data of this applicable change flag are "1", 23 is the N-ary updown counterS42, and a register file is changed. That is, the return instruction as a down enabling functional signal is inputted into the N-ary updown counter 23 for a change signal as a count-down functional signal, and, as a result, only -1 counts down the enumerated data to it. By this count-down of -1, a register file selection signal will choose the register file which it interrupted and was being used immediately before. Moreover, when the data of an applicable change flag are "0", count-down actuation of the above-mentioned N-ary updown counter 23 is not performed. And it moves to S44. [0035] To actuation and coincidence of the N-ary updown counter 23, 24 checks the contents of the change flag by the sequencerS41, and when the value is "1", a change flag is reset by S43. This is because it is necessary to set the value of a change flag to "0" and to give a status register 7 in order for the original register file to be chosen by count-down processing of the above-mentioned N-ary updown counter 23. For this reason, a change signal is inputted into a sequencer 24 as a recovery functional signal, and, as a result, the return instruction as a roll-back functional signal outputs the change flag of "0" to it. The example in this case is shown as that by which the change flag corresponding to RF3 on the right-hand side of drawing 4 is changed into "0" from "1." The condition of saying that it was rewound by what the register file interrupted and was being used immediately before by modification of the flag data in this status register 7 is shown.

[0036] Moreover, 24 checks the contents of the change flag, by the sequencerS41, when the value of a change flag is "0", interruption processing is usually performed, and since it is recognized as the change of a register file not having been carried out, modification processing of the above-mentioned flag data is not performed. And it moves to S44. So far, it is usually the difference in processing with the return from interruption processing with the return from the high-speed interruption processing in interruption return processing. In the case of both, the following becomes with the same actuation.

[0037] The interruption processing control section 9 is S44 of drawing 7, it gives a return signal to a status register 7, reads the contents of the status register 7 at the interruption time evacuated to the specific register of a register file through an internal bus, and returns to a status register 7. Next, nine is the interruption processing sectionS45, and a return signal is given to a program counter 1, the interrupted instruction address which is evacuated to the specific register of a register file is read, and it returns to a program counter 1. From a program counter 1, the instruction address of the beginning after a return is given to a memory apparatus 2, and the program execution interrupted by interruption is resumed.

[0038] In addition, since the change duration for the renewal of a register file performed when high-speed interruption processing is possible since each circuit in the register file selection section 11 of drawing 1 and the interrupt control section 9 of drawing 2 is constituted by the high-speed logical element, and a return is a short time very much and became unnecessary [ the evacuation to the memory apparatus of the contents of the conventional register file, and return processing ], beforehand [ of interruption processing ] and the duration of corrective action were reduced more sharply than before. Moreover, since the updated register file is used in high-speed interruption processing of this invention, there is no limit on the interruption processing program description produced when the usable numbers of registers are the case of processing in which it does not interrupt, and the same number and it uses a limited number in the conventional technique of registers in interruption processing.

[0039] Furthermore, in this invention, the operating condition of the JISUTA file for interruption processing is checked by hardware, and when controlling the register file change and a register file were changed, and when that is not right, the device in which a separate interrupt vector table is generated is given. For this reason, it becomes unnecessary to judge with software whether it is usable in a special register set at the time of a \*\*\*\*\* receptionist required in order to perform the change to the special register set for the interruption processing by the conventional method, and all the problems of the degradation by this judgment processing can be avoided.

[0040]

[Effect of the Invention] The interruption processor applied to this invention as mentioned above inputted two or more usual interrupt signals and two or more high-speed interrupt signals as the

register bank which consists of two or more register files, and the status register holding the status information concerning a processor, it has an interrupt control means to control branching and the return to each interruption processing program, and said interrupt control means usually has an interrupt control means and a high-speed interrupt control means. An interrupt control means usually, when [ said ] an interrupt signal is usually inputted Evacuation preservation of the maintenance information and the program return address of said status register at the time is carried out. A predetermined interruption processing program is branched from the interrupt vector table uniquely given by the applicable usual interrupt signal. It is made to return to the program under activation from the program return address which carried out evacuation preservation while returning the maintenance information on said status register which carried out evacuation preservation to the status register after activation termination of this branched interruption processing program before branching. When said high-speed interrupt signal is inputted, a high-speed interrupt control means Evacuation preservation of the maintenance information and the program return address of said status register at the time is carried out. next, in distinguishing the existence of the register file which is not used in said register bank and being The register file which is not used in a register bank instead of a register file in use at the time is assigned. A predetermined interruption processing program is branched from the interrupt vector table uniquely given by the applicable high-speed interrupt signal. It returns to the register file which was being used before branching after activation termination of this branched interruption processing program. It is made to return to the program under activation from the program return address which carried out evacuation preservation while returning the maintenance information on said status register which carried out evacuation preservation to the status register before branching. moreover, distinguish the existence of the register file which is not used in said register bank, and when there is nothing Since it was made to branch the interruption processing program specified by another interrupt vector table for making said control usually according to an interrupt control means perform, without performing new allotment of a register file When high-speed interruption processing is possible, change a register file in use by hardware at high speed at the time, and it branches to a high-speed interruption processing program. After this branched program execution termination, it can return to the register file which was being again used before branching at high speed, and can return to the program before branching. The evacuation processing to the memory apparatus of the contents of the register file and return processing become unnecessary like the conventional method, rather than before, prior processing and corrective action time amount of interruption processing are reduced sharply, and the interruption processing engine performance improves. Moreover, since the register file itself is changed in this invention at the time of high-speed interruption processing, the number of the registers contained in a register file does not have the limit on the interruption processing program description produced when performing interruption processing using a number of registers which a part of register file [ in / it is usually processing and the same number, consequently / the conventional method ] carried out by processing in which it does not interrupt was evacuated and returned, and were restricted.

---

[Translation done.]